Fiche TD8 : décidabilité et complexité

Exercice 1 (*) *Formalisation de problèmes*

Dans chacun des cas suivants, formaliser le problème de décision ou d'optimisation sous jacent à la description informelle donnée - si c'est possible.

1. Trouver le minimum d'un tableau d'entiers.
2. Calculer la composante fortement connexe d'un sommet dans un graphe orienté.
3. Trier un tableau en place.
4. Le problème SAT.
5. Etant donné des objets ayant chacun un poids et une valeur, remplir un sac à dos ayant le plus de valeur possible tout en ne dépassant pas un poids donné.
6. Le problème d'accessibilité dans un graphe.

Exercice 2 (**) *Variantes autour du problème de l'arrêt*

Pour chacun des problèmes suivants, déterminer en justifiant s'il est décidable ou non. Dans les cas où le problème concerné est indécidable, on s'aidera d'une réduction depuis le problème de l'arrêt.

1. **Entrée** : Une fonction f, un argument x et un entier n.
   Question : Est-ce que l'exécution de f sur l'entrée x termine en moins de n opérations élémentaires ?
2. **Entrée** : Une fonction f et un argument x.
   Question : Est ce que $f(x)$ renvoie 0 ?
3. **Entrée** : Une fonction f.
   Question : Est-ce que l'exécution de $f(f)$ (f appliquée à son propre code) termine en temps fini ?
4. Dans cette question, f est une fonction fixée.
   Entrée : Un argument x.
   Sortie : Est-ce que $f(x)$ termine en temps fini ?
5. **Entrée** : Une fonction f.
   Question : Existe-t-il une entrée x telle que l'exécution de $f(x)$ termine en temps fini ?

Bonus (***) : On note $\mathsf{ARRET}_\exists$ le problème de décision de la question 5. Montrer que ce problème est semi-décidable. En déduire que co$\mathsf{ARRET}_\exists$ ne l'est pas.

Exercice 3 (***) *Problème de correspondance de Post*

Le problème de correspondance de Post, PCP, est le problème de décision suivant :

Entrée : Une suite finie $(u_1, v_1), ..., (u_n, v_n)$ de couples dont les éléments sont des mots sur un alphabet Σ.
Question : Existe-t-il un entier k et une suite d'indices $i_1, ..., i_k \in [\![1, n]\!]$ tels que $u_{i_1}...u_{i_k} = v_{i_1}...v_{i_k}$?

Le problème de correspondance de Post marqué, MPCP, lui est très similaire :

Entrée : Une suite finie $(u_1, v_1), ..., (u_n, v_n)$ de couples dont les éléments sont des mots sur un alphabet Σ.
Question : Existe-t-il un entier k et une suite d'indices $i_2, ..., i_k \in [\![1, n]\!]$ tels que $u_1 u_{i_2}...u_{i_k} = v_1 v_{i_2}...v_{i_k}$?

Ces problèmes peuvent être visualisés comme suit : leur entrée est un set de dominos verticaux $\boxed{\begin{array}{c} u_i \\ \hline v_i \end{array}}$ et l'objectif est de savoir s'il existe un enchaînement de dominos (sachant qu'on peut utiliser plusieurs fois un même domino) tels que le mot formé par la concaténation des parties hautes des dominos soit égal à celui formé par les parties basses. Dans le problème MPCP, on exige de plus qu'une telle suite commence par un domino spécifique : le premier.

1. La suite $(a, baa), (ab, aa), (bba, bb)$ est-elle une instance positive de PCP ?
2. Montrer que $\mathsf{MPCP} \leq \mathsf{PCP}$.

3. En déduire que PCP est décidable si et seulement si MPCP l'est.

On admet que MPCP est indécidable. Ce résultat va nous servir à montrer l'indécidabilité de problèmes de décision portant sur les langages algébriques. Si $(u_1, v_1), ..., (u_n, v_n)$ est une instance de PCP dans laquelle les lettres des mots appartiennent à un alphabet Σ, on peut toujours introduire un alphabet $A = \{a_1, ..., a_n\}$ formé de lettres n'appartenant pas à Σ et on note alors L_u le langage :

$$L_u = \{u_{i_1}...u_{i_k} a_{i_k}...a_{i_1} \mid k \geq 0 \text{ et } 1 \leq i_r \leq n\}$$

4. Montrer que L_u est algébrique.
5. Montrer l'indécidabilité du problème INTERSECTION_VIDE :

   Entrée : Deux grammaires algébriques G et G'.
   Question : L'intersection des langages engendrés par G et G' est-elle vide ?

6. En déduire l'indécidabilité du problème AMBIGUITE :

   Entrée : Une grammaire algébrique G.
   Question : G est-elle ambiguë ?

Exercice 4 (*) *Classe de complexité a priori*

Dans chacun des cas, indiquer la classe de complexité déterministe du problème (a priori) en justifiant :

1. **Entrée** : Trois entiers naturels a, b, k.
   Question : Le k-ème bit dans l'écriture bianire de $a \times b$ vaut-il 1 ?
2. **Entrée** : Un graphe $G = (S, A)$, deux sommets $s, t \in S$ et un entier k.
   Question : Y a-t-il un chemin de s à t de longueur au plus k ?
3. **Entrée** : Un graphe $G = (S, A)$, deux sommets $s, t \in S$ et un entier k.
   Question : Y a-t-il un chemin de s à t de longueur au moins k ?
4. **Entrée** : Des poids $p_1, ..., p_n \in \mathbb{N}$, un nombre de boîtes $m \in \mathbb{N}$ et un poids maximal $P \in \mathbb{N}$.
   Question : Peut-on ranger n objets de poids $p_1, ..., p_n$ dans m boîtes sans que le poids d'aucune ne dépasse P ?
5. **Entrée** : Une fonction f et un tableau t de taille n.
   Question : Le calcul de $f(t)$ termine-t-il en moins de n étapes de calcul ?
6. **Entrée** : Un graphe non orienté $G = (S, A)$ et un entier k.
   Question : Existe-t-il dans G un ensemble $B \subset A$ tel que $|B| \leq k$ et tel que pour tout sommet s de G au moins une arête de B est incidente à s ?
7. Trouver l'élément minimal dans un arbre binaire.
8. Déterminer si un graphe non orienté est un arbre.
9. Calculer un arbre couvrant minimal dans un graphe pondéré connexe et non orienté.
10. Savoir s'il existe un couplage dans un graphe biparti.

Exercice 5 (*) *Problèmes NP*

Montrer que les problèmes suivants sont dans NP en exhibant un certificat et un vérificateur polynomial.

1. SUBSET SUM: $\begin{cases} \textbf{Entrée} : \text{Un ensemble fini d'entiers naturels } E \text{ et une cible } C \in \mathbb{N}. \\ \textbf{Question} : \text{Existe-t-il un sous ensemble de } E \text{ dont la somme des éléments vaut } C \text{ ?} \end{cases}$
2. RANGEMENT est le problème décrit en question 4 de l'exercice 4.
3. SAC A DOS : $\begin{cases} \textbf{Entrée} : \text{Une liste de poids } p_1, ..., p_n \in \mathbb{N}, \text{ une liste de valeurs } v_1, ..., v_n \in \mathbb{N}, \\ \qquad\qquad \text{un poids maximal } P \in \mathbb{N} \text{ et une valeur minimale } V \in \mathbb{N}. \\ \textbf{Question} : \text{Existe-t-il un sous ensemble } I \subset [\![1, n]\!] \text{ tel que } \sum\limits_{i \in I} p_i \leq P \text{ et } \sum\limits_{i \in I} v_i \geq V \text{ ?} \end{cases}$

4. CHEMIN EULERIEN : $\begin{cases} \textbf{Entrée} : \text{Un graphe } G=(S,A). \\ \textbf{Question} : \text{Existe-t-il un chemin qui passe une et une seule fois par chaque arête de } A \text{ ?} \end{cases}$

5. COUPLAGE PARFAIT : $\begin{cases} \textbf{Entrée} : \text{Un graphe } G=(S,A) \text{ non orienté.} \\ \textbf{Question} : \text{Existe-t-il un couplage parfait dans } G \text{ ?} \end{cases}$

6. COUPE : $\begin{cases} \textbf{Entrée} : \text{Un graphe } G=(S,A) \text{ et un entier } k. \\ \textbf{Question} : \text{Existe-t-il une partition } X \sqcup Y \text{ de } S \text{ telle qu'au moins } k \text{ arêtes relient } X \text{ à } Y \text{ ?} \end{cases}$

Remarque : Certains de ces problèmes sont en fait dans P, lesquels ?

Exercice 6 (**) *Voyageur de commerce*

On considère les deux problèmes de décision suivants :

CYCLE HAMILTONIEN:

Entrée : Un graphe non orienté $G = (S, A)$.

Question : Existe-t-il dans G un cycle hamiltonien (cad passant une et une seule fois par chaque sommet) ?

VOYAGEUR DE COMMERCE:

Entrée : Un graphe non orienté $G = (S, A)$ pondéré par p et un entier naturel k.

Question : Existe-t-il dans G un cycle hamiltonien de poids inférieur ou égal à k ?

1. En utilisant le fait que CHO (cycle hamiltonien orienté) est NP-complet, montrer via une réduction polynomiale bien choisie que CYCLE HAMILTONIEN est aussi NP-complet.
2. Montrer que VOYAGEUR DE COMMERCE est NP-complet.

Exercice 7 (**) *Réductions autour du problème du sac à dos*

Dans cet exercice, on considère les problèmes SUBSET SUM, SAC A DOS et RANGEMENT, définis à l'exercice 5 ainsi que le problème PARTITION suivant :

Entrée : Une liste d'entiers $a_1, ..., a_n$.
Question : Existe-t-il un sous ensemble $I \subset [\![1, n]\!]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$?

1. Montrer que SUBSET SUM $\leq$ SAC A DOS.
2. Montrer que SUBSET SUM $\leq$ PARTITION.
3. Montrer que PARTITION $\leq$ RANGEMENT.
4. En admettant que le problème SUBSET SUM est NP-complet (ce résultat sera montré à l'exercice 9), que peut-on en déduire sur la classe de complexité des problèmes de cet exercice ?

Exercice 8 (***) *Systèmes d'équations booléennes*

1. On considère le problème consistant à déterminer si un système d'équations linéaires booléen (c'est-à-dire dont les inconnues sont à valeurs dans $\{0, 1\}$) admet une solution. Montrer que ce problème est dans P.
2. Que peut-on dire de la classe de complexité de ce même problème si les équations sont polynomiales plutôt que linéaires ?

Exercice 9 (***) *Subset sum*

L'objectif de cet exercice est de montrer que le problème

SUBSET SUM : $\begin{cases} \textbf{Entrée} : \text{Un tableau } T = [t_1, ..., t_n] \text{ d'entiers naturels et } C \in \mathbb{N}. \\ \textbf{Question} : \text{Existe-t-il un sous ensemble } I \subset [\![1, n]\!] \text{ tel que } \sum_{i \in I} t_i = C \text{ ?} \end{cases}$

est NP-complet. On a déjà montré à l'exercice 5 qu'il était NP. Pour montrer qu'il est NP-difficile, on procède via une réduction depuis 3SAT. Soit φ une formule sous 3-CNF dont les variables sont $x_1, ..., x_n$ et les clauses $C_0, ..., C_{m-1}$. Pour tout $i \in [\![1, n]\!]$, on définit a_i et b_i par :

$$a_i = 6^{i+m-1} + \sum_{\substack{j=0 \\ x_i \in C_j}}^{m-1} 6^j \text{ et } b_i = 6^{i+m-1} + \sum_{\substack{j=0 \\ \overline{x_i} \in C_j}}^{m-1} 6^j$$

De plus, pour tout $j \in [\![0, m-1]\!]$, on pose $c_j = d_j = 6^j$ et on note $C = 3 \sum_{i=0}^{m-1} 6^i + \sum_{j=m}^{m+n-1} 6^j$.

1. Considérons tout d'abord la formule exemple $\varphi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$. Dans un tableau à 15 lignes et 7 colonnes donner la représentation en base 6 des a_i, b_i, c_j, d_j et de C. La colonne de droite sera celle des "bits" de poids faible.
2. Toujours dans ce cas particulier, si on ne prend pas en compte la ligne représentant la décomposition de C, pour chacune des quatre premières colonnes, combien de lignes au maximum dans le tableau précédent contiennent un 1 dans celle-ci ? Même question pour les trois dernières colonnes.

On pose $T = [a_1, ..., a_n, b_1, ...b_n, c_0, ...c_{m-1}, d_0, ...d_{m-1}]$.

3. On suppose que φ est satisfiable. Montrer que (t, C) est une instance positive de SUBSET SUM.
4. Réciproquement, montrer que si (t, C) est solution de SUBSET SUM alors φ est satisfiable.
5. En déduire que SUBSET SUM est NP-complet.

Si $([t_1, ..., t_n], C)$ est une instance de SUBSET SUM, pour tout $i \in [\![0, n]\!]$ et $c \in [\![0, C]\!]$, on note $SP(i, c)$ le booléen valant vrai s'il existe un sous ensemble $I \subset [\![1, i]\!]$ tel que $\sum_{j \in I} t_j = C$ et faux sinon.

6. Déterminer les valeurs de $SP(i, c)$ lorsque $c = 0$, $c < 0$ ou $i = 0$.
7. Déterminer une formule de récurrence liant $SP(i, c)$ à $SP(i-1, c)$ et $SP(i-1, c-t_i)$.
8. En déduire un algorithme permettant de résoudre SOMME PARTIELLE avec une complexité en $O(nC)$.
9. Vient-on de montrer que P = NP ?